AdaBoosting Graded Homework

The first part of this assignment is (almost) identical to the example of AdaBoosting that was discussed in class.

Input

A file containing the following information:

```
T (an integer number).
n (an integer number).
epsilon (a small real number)
x (a list of n real numbers. These are assumed to be in increasing order).
y (a list of n numbers, each one is either 1 or -1).
p (a list of n nonnegative numbers that sum up to 1).
```

Example:

```
10 4 0.0000001
1 2 3.5 4.5
1 -1 1 1
0.25 0.25 0.25 0.25
```

Weak classifiers

The weak classifier produces hypotheses of the form: $x < v$, or $x > v$. It is always computed from the entire data. (No sampling.)

Part I. Binary AdaBoosting

(The value of epsilon given in the first input line is ignored here.)

What should be computed

Run T iterations of the binary AdaBoosting algorithm. For each iteration compute and print the following:

1. The selected weak classifier: h_t.

2. The error of h_t: ϵ_t.

3. The weight of h_t: α_t.

4. The probabilities normalization factor: Z_t.

5. The probabilities after normalization: p_i.

6. The boosted classifier: f_t.

7. The error of the boosted classifier: E_t.

8. The bound on E_t.

$$\prod_{j=1}^{t} Z_t$$

Part II. Real AdaBoosting

The value of epsilon given in the first input line is used to smooth the classifiers.

What should be computed

Run T iterations of the real AdaBoosting algorithm. For each iteration compute and print the following:

1. The selected weak classifier: h_t.

2. The G error value of h_t.

3. The weights c_t^+, c_t^-.

4. The probabilities normalization factor: Z_t.

5. The probabilities after normalization: p_i.

6. The values $f_t(x_i)$ for each one of the examples.

7. The error of the boosted classifier: E_t.

8. The bound on E_t.

$$\prod_{j=1}^{t} Z_t$$

What you need to submit

Submit source code and executable of two programs. You can submit it online (on WebCT) or on a diskette, CD, etc. Make sure that your submission includes both source code and binaries. If you use a UTD Unix machine add a printout of the command "ls -l project-directory", and make sure that all programs have read and executable permission. Also, be careful not to change the size and date of the files in that directory until you get your project back.
You must be available to demonstrate your program to the TA. Dates will be announced in class.